CLAIMS

What is claimed is:

1. A computer implemented method for communicating data in a clustered computing system, the method comprising:

detecting an occurrence of a first event at a first node of the system;

detecting an occurrence of a second event at the first node of the system;

determining that

the first event is identical to the second event, and

the occurrence of first event and the occurrence of the second event are therefore a first occurrence and a second occurrence of an identical event;

appending a notification including the information describing the identical event onto an existing message, wherein

the notification is appended only once for both the first occurrence and the second occurrence,

the notification is not appended twice, once for each occurrence, and

the message was destined to be propagated to a second node; and

propagating the message along with the information describing the identical event to the second node.

2. The method of claim 1 wherein the message was generated for purposes other than sending information appended.

3. The method of Claim 1, wherein the determining further comprises

comparing information that describes the first event with information that describes the second event to determine whether the first event and the second event are identical; and

the method further comprising if the two events are identical, then indicating that the information that describes the second event no longer needs to be retained.

4. The method of Claim 1 wherein the method further comprises:

setting an identifier indicating that the information describing the identical event is to be appended onto a message and propagated to a particular node.

5. The method of Claim 1, wherein said clustered computing system comprises a database management system.

6. The computing environment of Claim 1, wherein said clustered computing system comprises a shared-disk database system.

7. The computing environment of Claim 1, wherein said clustered computing system comprises a shared-cache parallel database management system.

8. The computing environment of Claim 1, wherein said clustered computing system comprises a shared-nothing database management system.

9. The computing environment of Claim 1, wherein said clustered computing system comprises a distributed database management system.

10. The method of Claim 1, wherein the method further comprises:

searching a shared-memory event buffer having a size that is fixed.

11. The method of Claim 1 wherein the message has a fixed size, and the method further comprises:

appending additional information that describes additional events onto existing message traffic until free space in the fixed-size message is filled.

12. The method of Claim 1, wherein the method further comprises

placing the information describing the identical event in a queue.

13. The method of claim 12, wherein the queue includes at least a priority queuing mechanism in order to determine a priority for events such that high priority events would supercede a low priority events in an event notification queue.

14. The method of claim 1 wherein an in-memory hash index is used to determine if an event exists in a shared-memory event buffer.

15. The method of claim 1 wherein the method further comprises:

partitioning a shared-memory event buffer;

generating an event buffer entry of the shared memory event buffer;

placing an event identifier into the event buffer entry; and

inserting the information describing the identical event into the event buffer entry.

16. The method of claim 15 the method further comprises if between a fastest head pointer and a tail pointer there does not exists a buffer entry in the shared memory event buffer for the identical event, generating a new event buffer entry, and the inserting further

comprises inserting the information describing the identical event into the new event buffer entry.

17. The method of claim 15 wherein the inserting comprises:

if between a fastest head pointer and a tail pointer there exists a buffer entry in the shared memory event buffer for the identical event, updating the buffer entry so that the buffer entry represents the second occurrence.

18. The method of claim 15 further comprising

using a round robin method and the shared memory event buffer to determine to which existing message to appended the information describing the identical event.

19. The method of Claim 1, wherein the method further comprises the step of:

maintaining the information that describes a plurality of events.

20. The method of Claim 19, wherein the method further comprises the step of :

maintaining the information that describes the plurality of events in a shared-memory event buffer .

21. The method of Claim 19, wherein the method further comprises:

maintaining the information that describes the plurality of events in a circular buffer.

22. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

23. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

24. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

25. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

26. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

27. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

28. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

29. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

30. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

31. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.

32. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.

33. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.

34. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.

35. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 14.

36. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 15.

37. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 16.

38. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 17.

39. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 18.

40. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 19.

41. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 20.

42. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 21.

43. A computer implemented method for communicating data in a clustered computing system, the method comprising:

detecting a generation of an event at a first node of the system, determining if the event is identical to another event that was previously generated,

appending only one copy of the information is appended onto an existing message whether or not the other event is identical, wherein the message was destined to be propagated to a second node; and

propagating the message along with the information describing the identical event to the second node.

44. The method of claim 43, wherein the determining further comprises:

if

there exists a stored indication that an identical event was previously generated and

the propagating of the message having the information appended did not yet occur,

then an indication is stored that multiple identical events were generated.

45. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 43

46. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 44.

47. A computer implemented method comprising the computer-implemented steps of:

receiving, at a database server that is executing on a second node in a distributed system, a message that was transmitted by a first node in the distributed system wherein the message has appended thereon information that describes an event that occurred at least a second time before the message was sent, and

a second description of the event is not received despite the second occurrence; and

retrieving the information that describes the event signaling processes in the local node that said event has occurred.

48. The method of claim 47, wherein the method further comprises:

invoking concurrency control techniques to control concurrent access to a shared-memory event buffer from processes that propagate messages to subscriber nodes and processes that generate events.

49. The method of claim 47, wherein said method further comprises:

maintaining information that describes a plurality of events,

coalescing the information that describes a plurality of events, wherein the information that describes a plurality of events may be coalesced for the same event into a single event notification.

50. The method of Claim 47, wherein the method further comprises:

maintaining the information that describes a plurality of events.

51. The method of Claim 47, wherein the method further comprises step of:

maintaining the information that describes the plurality of events in a shared-memory event buffer .

52. The method of Claim 47, wherein the method further comprises the step of:

maintaining the information that describes the plurality of events in a circular buffer.

53. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 47.

54. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 48.

55. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 49.

56. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 50.

57. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 51.

58. A computer-readable medium carrying one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 52.

59. A computer apparatus comprising:

a processor; and

a memory coupled to the processor, the memory containing one or more sequences of instructions for event notification in a clustered computing environment, wherein execution of the one or more sequences of instructions by the processor causes the processor to perform the method of claim 1.